(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 246 072 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.10.2002 Bulletin 2002/40

(51) Int Cl.7: G06F 13/40

(21) Application number: 02252252.8

(22) Date of filing: 27.03.2002

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 27.03.2001 US 818951

(71) Applicant: INTEL CORPORATION
Santa Clara, CA 95052 (US)

(72) Inventor: Solomon, Gary A.
Acton, MA 01720 (US)

(74) Representative: Loveless, Ian Mark
Reddie & Grose,
16 Theobalds Road
London WC1X 8PL (GB)

(54) Adaptive read pre-fetch

(57) An amount of data to be pre-fetched during read operations is adaptively modified based upon the experience of previous reads. If previous reads were terminated before all the data desired was obtained, subsequent read amounts may be increased. The initial amount of pre-fetched data may be pre-set or modified dynamically.

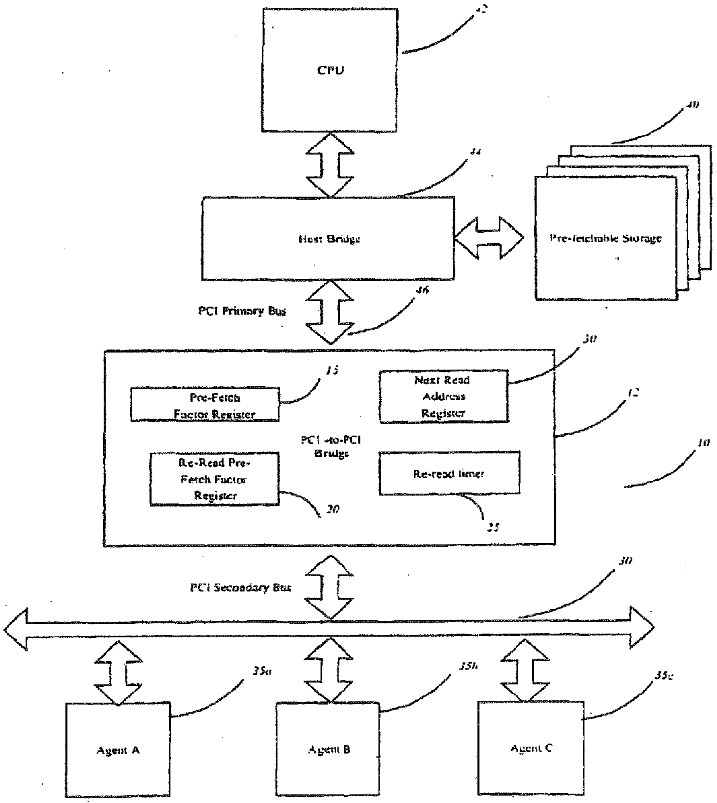

FIG 1

Printed by Jouve, 75001 PARIS (FR)

Description

FIELD OF THE INVENTION

[0001] This invention relates to computer read pre-fetch operations,.

BACKGROUND OF THE INVENTION

[0002] . Existing PCI bridges assist in the control of the sequencing of operations and access to computer busses in accordance with the bus specification (such as, for example, PCI Local Bus Specification Rev. 2.2 published by the PCI Special Interest Group). Pre-fetch algorithms are not covered by the PCI specification, but are widely employed by FCI devices to circumvent a fundamental issue with PCI protocol: it does not include a read amount embedded within each transaction. Such devices employ a static read pre-fetch which requests the same amount of information for a particular type of read operation, regardless of the actual demands of the requesting agent. While this constant pre-fetch amount may be adjustable by means of a device specific configuration register, the selected amount is constant and applicable to all requesting agents served in connection with that register. A static pre-fetch amount may result in pre-fetching too much data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003]

FIG. 1 is a block diagram of an adaptive read pre-fetch system.
FIG 2. is a flow chart of an adaptive pre-fetch read method.

DETAILED DESCRIPTION

[0004] Referring to FIG 1, an example adaptive adaptive read pre-fetch system 10 is shown having components on a bridge 12. The components include a pre-fetch factor register 15, being a re-writeable storage location. The adaptive read pre-fetch system 10 also includes a re-read pre-fetch factor register 20, a re-read timer 25 and a next read address register 30. Also shown is pre-fetchable data storage such as system memory 40, and agents 35a, 35b and 35c. Each of the components of the adaptive read pre-fetch system 10 are preferably part of or attached to the computer, such as a bridge 12, within which the pre-fetch factor register 15, the re-read pre-fetch factor register 20, the re-read timer 25, and the next read address register 30 may, but need not, reside. Also shown in Fig. 1 is a CPU 42 which communicates through a host bridge 44 with a PCI primary bus 46. Bridge 12 is also capable of communicating with the primary bus 46.

[0005] An agent 35a, 35b or 35c may be any requesting agent, such as an agent on a PCI 2.2 secondary bus 30 connected to a bridge 12. An agent may be any of a number of devices capable of requesting a memory read operation on the bus.

[0006] At a set time, typically upon system reset, the values in the pre-fetch factor register 15 and re-read pre-fetch factor register 20 are initialized.

[0007] When an agent on the bus 30 requests a memory read operation, it notifies bridge 12 of the request by asserting the appropriate signals on the bus 30. If the bridge 12 determines that the request is from pre-fetchable storage 40, it multiplies a pre-defined amount of data requested by the number held in the pre-fetch factor register 15. The amount of data to be read depends upon the type of read request as well as the particular system design, for example the size of a cache line. Table 1 shows the data amounts for three types of read requests. PFFR is the pre-fetch factor register value.

Memory Operation	Alignment	Read Size
Read	DWORD	(PPFR+1)*4*DWORD
Read Line	Cacheline	(PPFR+1)*cacheline
Read multiple	2 cachelines	(PPFR+1)*2 cachelines

[0008] A cacheline is a series of contiguous bytes of data corresponding to the host CPU's cache subsystem. Cachelines conform to CPU dependent address alignment. A DWORD is a double word, with a length that depends upon the particular computer memory configuration. Read operations may be limited to cacheline boundaries. Factor is the value contained in the pre-fetch factor register, and may be altered during operation of the computer by software.

[0009] Referring to FIG 2, a flow chart of an adaptive read pre-fetch method 100 is shown. At system initialization 105, an initial value for the pre-fetch factor register is set. This may be in system ROM, or may be set (and changed from time to time) as a parameter by the operating system or any other system or application software. In one embodiment, pre-fetch timer may be initialized to a set time, which will decrement to zero unless reset.

[0010] If an agent gives a pre-fetchable read request 110 (of whatever type) then the read amount, based upon the type of read, (see table 1) is multiplied by the pre-fetch factor plus one, the pre-fetch factor being stored in the pre-fetch factor register 15 . Thus, if the value of the pre-fetch factor register is zero, the read amount is multiplied by one, effectively disabling the feature.

[0011] The value in the next read register 30 is compared to the value of the read address received from the agent. If they are the same (meaning that the value in the next read address was stored as a result of a prior read request from the same agent which was terminated early for some reason, such as being disconnected by the bridge for lack of data), then the read amount is again increased. The read amount is multiplied by one plus the value in the re-read pre-fetch factor register 20. Other implementations could successively automatically increase the value in the re-read pre-fetch factor register for each early terminated read, and conversely could periodically decrement the re-read pre-fetch factor.

[0012] If the address in the read request does not match the value in the next-read address 125, the value in the re-read pre-fetch register is ignored. In either case, the calculated pre-fetch amount is attempted to be read 135.

[0013] Table 2 shows the read size for different memory operations using the re-read pre-fetch register (RRPFR) value:

Table 2

Memory Operation	Alignment	Read Size
Read	DWORD	(PPFR+1+RRPFR)*4*DWORD
Read Line	cacheline	(PPFR+1+RRPFR)*cacheline
Read multiple	2 cachelines	(PPFR+1+RRPFR)*2cachelines

[0014] If the read terminates early, then the requesting agent has not received all of the data that presumably it presumably wants. Early termination occurs if the bridge disconnects the read transaction because data is exhausted and the requesting device is still expecting additional data (i.e. still asserting the PCI bus signal FRAMED.) Data may become exhausted because of a variety of reasons, including an end of file, exhaustion of a buffer or other causes.

[0015] In the case of a first early termination, the adaptive read pre-fetch process increases the amount of data retrieved on the next read request at the same location (where the current read ended) from the requesting agent. This is accomplished by saving the next-read address (the next address at which data would have been retrieved had the read not been terminated early) and beginning to use the re-read factor and command type specific pre-fetch amounts.

[0016] The smart pre-fetch ability may be disabled by programming that is accessible during system initialization and by the operating system as a parameter. A separate process may be implemented for each agent on a secondary bus and may also be implemented in the primary bus side as well as the secondary bus.

[0017] The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, the process may be implemented on a bridge, a separate circuit (discrete or integrated) or in software, or in combinations of software and firmware or circuitry. It may be used successfully in other than a PCI 2.2 bus system. Not all parts of the described embodiment need be implemented to achieve beneficial results.

Claims

1. A method comprising:

   receiving, from an agent, a request to read data from a read address in pre-fetchable data storage;
   retrieving an initial amount of data determined by a pre-fetch factor;
   determining if the requesting agent received as much data as it requested; and
   based upon the determining, storing a next read address.

2. The method of claim 1 further comprising, before the retrieving step, comparing the read address to a stored next read address, and if they match, retrieving an amount of data determined by both the pre-fetch factor and a re-

read pre-fetch factor.

3. The method of claim 2 further comprising changing the re-read pre-fetch factor based upon the determining.

4. The method of claim 2 or 3 further comprising the step of changing the re-read pre-fetch factor after an interval.

5. The method of any preceding claim wherein the value of the pre-fetch factor is alterable.

6. The method of claim 3, wherein the changing is selectively enabled and disabled.

7. A system comprising:

a computer having at least one agent, at least one bridge, a pre-fetch factor register, a re-read pre-fetch factor register and a next read address register;
the bridge being configured to

(a) receive from an agent a request to read data from a read address in pre-fetchable data storage;
(b) request an amount of data determined by a number stored in the pre-fetch factor register;
(c) determine if the requesting agent has received the full amount of requested data;
(d) based upon the determination, increment the re-read pre-fetch factor register.

8. The system of claim 8 wherein the bridge is further configured, based upon the determination, to store a next read address.

9. The system of claim 7 or 8 further comprising the bridge being configured to compare the read address to the stored next read address, and if they match, changing the amount of data determined also by the value in the re-read pre-fetch factor register.

10. The system of claim 7, 8 or 9 further comprising the bridge further being configured to change the value in the re-read pre-fetch factor register based upon the determining.

11. The system of any of claims 7 to 10 further comprising the bridge further being configured to decrement the pre-fetch factor register after an interval.

12. The system of any of claims 7 to 11 wherein the contents of the pre-fetch factor register is alterable.

13. The system of any of claims 7 to 11 wherein the bridge is further configured so as to be able to enable and disable the application of the pre-fetch register and the re-read pre-fetch register under control of the computer.

14. The system of any of claims 7 to 13 wherein the pre-fetch register is contained within the bridge.

15. The system of any of claims 7 to 14 wherein the re-read pre-fetch register is contained within the bridge.

16. A computer program product, disposed on a computer readable medium, comprising instructions to cause a computer to undertake the method of any of claims 1 to 7 or to create the system of any of claims 8 to 15.

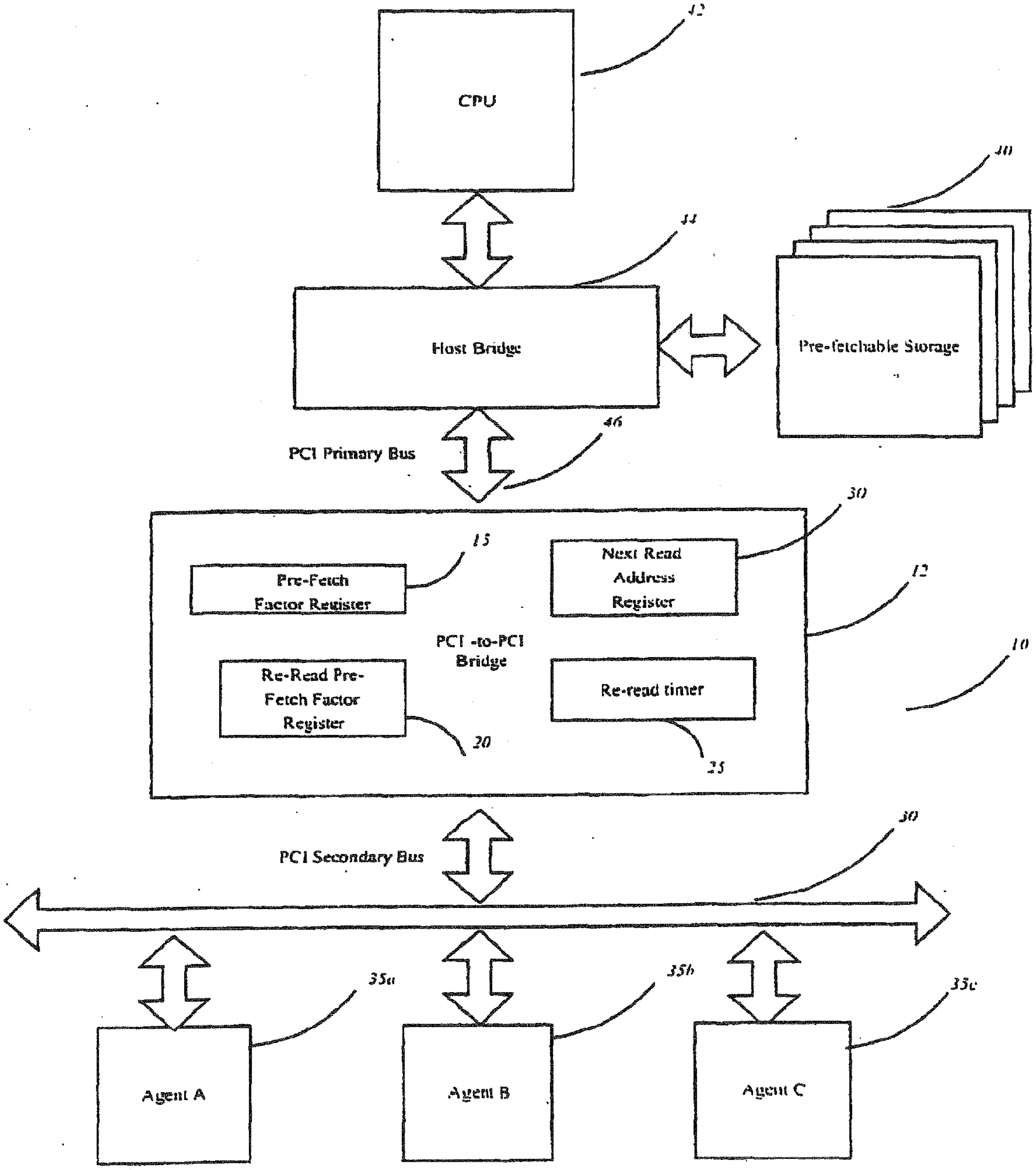
CPU
42
40
44
Host Bridge
Pre-fetchable Storage
PCI Primary Bus
46
30
15
Pre-Fetch Factor Register
Next Read Address Register
12
PCI -to-PCI Bridge
10
Re-Read Pre-Fetch Factor Register
Re-read timer
20
25
30
PCI Secondary Bus
35a
35b
35c
Agent A
Agent B
Agent C

FIG 1

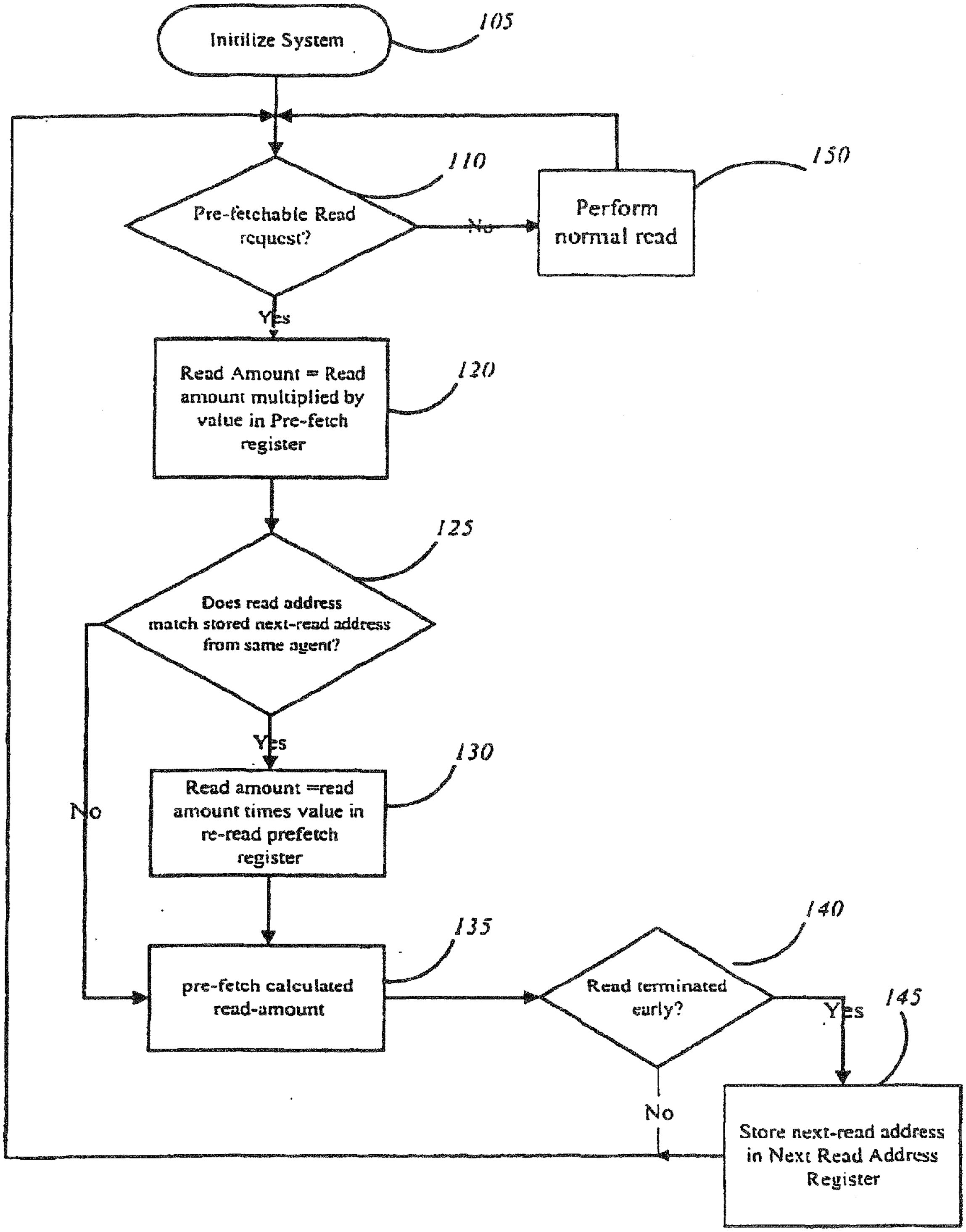
Initilize System
105
Pre-fetchable Read request?
110
No
Perform normal read
150
Yes
Read Amount = Read amount multiplied by value in Pre-fetch register
120
Does read address match stored next-read address from same agent?
125
Yes
No
Read amount =read amount times value in re-read prefetch register
130
pre-fetch calculated read-amount
135
Read terminated early?
140
Yes
145
No
Store next-read address in Next Read Address Register

FIG. 2

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 25 2252

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	US 6 012 106 A (SCHUMANN REINHARD C ET AL) 4 January 2000 (2000-01-04) * column 1, line 29 - line 35 * * column 2, line 8 - line 12 *	1-16	G06F13/40
A	US 5 761 464 A (HOPKINS CHARLES H) 2 June 1998 (1998-06-02) * abstract * * column 1, line 24 - line 38 * * column 1, line 50 - line 57 * * column 2, line 9 - line 16 *	1-16	
A	US 5 768 548 A (YOUNG BRUCE ET AL) 16 June 1998 (1998-06-16) * abstract * * column 8, line 22 - column 9, line 48 *	1-16	
A	EP 0 924 620 A (COMPAQ COMPUTER CORP) 23 June 1999 (1999-06-23) * abstract * * paragraphs '0007!,'0034!,'0035!,'0061!; figure 9 *	1-16	TECHNICAL FIELDS SEARCHED (Int.Cl.7) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	10 June 2002	Albert, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 02 25 2252

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

10-06-2002

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 6012106	A	04-01-2000	NONE		
US 5761464	A	02-06-1998	NONE		
US 5768548	A	16-06-1998	NONE		
EP 0924620	A	23-06-1999	EP	0924620 A2	23-06-1999
			JP	11316707 A	16-11-1999

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82